EE 615
Experiment No. 3

Aero 2-DoF: PID Controller for Pitch and Yaw Control

Lab report submitted by

Group 3 and 12

Aatmaj Barbhaiya: 24M1092
Aaditya Nagare: 21D070001
Drishtant Jain: 24M1085
Gagan G: 24M1090

March 11, 2025

Under the guidance of

Prof. Debasattam Pal

Department of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

Contents

1 Introduction

The Aero 2 Degree of Freedom (DoF) system is a widely used setup to understand the control dynamics of aerospace vehicles. It enables the study of pitch and yaw movements and the implementation of control strategies to achieve desired performance. This experiment involves designing and implementing PID controllers in SIMULINK to regulate the pitch and yaw movements of the system.

2 Aim

The aim of this experiment is to design and implement two PID controllers in **SIMULINK** to separately control the **pitch** and **yaw** of the Aero 2-DOF system.

3 Objectives

- Analyze the dynamics of the Aero 2-DOF system and derive its mathematical model.
- Design and implement PID controllers for independent pitch and yaw control.
- Tune the controller gains to achieve the required system performance:
  - Peak time: $t_p \leq 2.5s$ for pitch and $t_p \leq 3.5s$ for yaw.
  - Percent Overshoot: $PO \leq 5\%$ for both axes.
- Validate the system response through simulations and real-time hardware implementation.

4 Control Algorithm

The **control algorithm flowchart** is as follows:

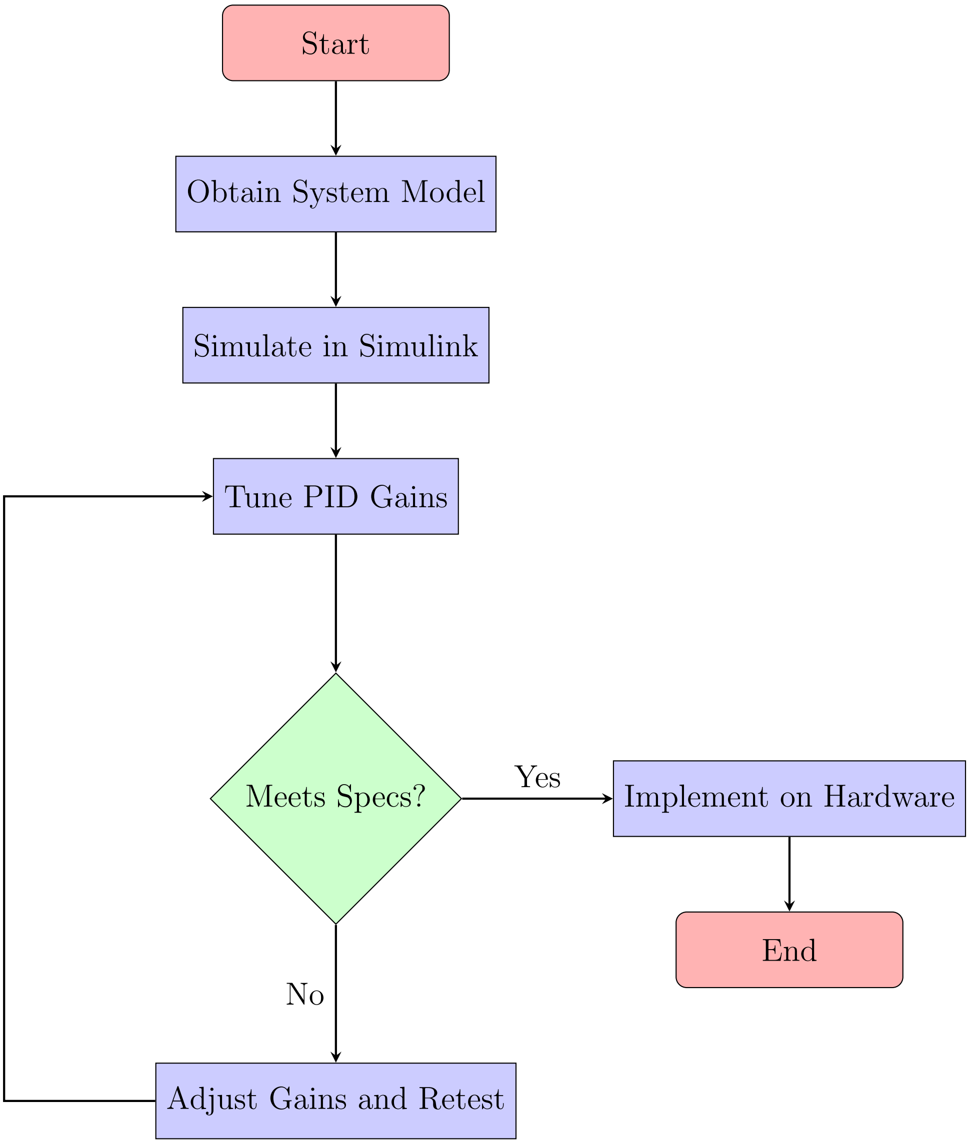

Figure 1: Control Algorithm Flowchart

5 Mathematical Model and Transfer Function

The equations of motion governing the system are:

$$J_p\ddot{\theta} + D_p\dot{\theta} = K_{pp}V_pD_t + K_{py}V_yD_t \tag{1}$$

$$J_y\ddot{\psi} + D_y\dot{\psi} = K_{yp}V_pD_t + K_{yy}V_yD_t \tag{2}$$

Taking the Laplace transform of the equations:

$$J_ps^2\Theta(s) + D_ps\Theta(s) = K_{pp}V_pD_t + K_{py}V_yD_t \tag{3}$$

$$J_ys^2\Psi(s) + D_ys\Psi(s) = K_{yp}V_pD_t + K_{yy}V_yD_t \tag{4}$$

Rearranging for the transfer functions:

$$\Theta(s) = \frac{K_{pp}D_t}{J_ps^2 + D_ps + K_{sp}}V_p + \frac{K_{py}D_t}{J_ps^2 + D_ps + K_{sp}}V_y \tag{5}$$

$$\Psi(s) = \frac{K_{yp}D_t}{J_ys^2 + D_ys}V_p + \frac{K_{yy}D_t}{J_ys^2 + D_ys}V_y \tag{6}$$

Given the parameter values:

- $J_p = 0.0232$ (total moment of inertia about pitch axis)
- $D_p = 0.0020$ (damping about pitch axis)
- $K_{sp} = 0.0074$ (stiffness about pitch axis)
- $J_y = 0.0238$ (total moment of inertia about yaw axis)
- $D_y = 0.0019$ (damping about yaw axis)
- $K_{pp} = 0.0032$ (thrust force gain acting on the pitch axis from the pitch/front rotor)
- $K_{py} = 0.0014$ (thrust force gain acting on the pitch axis from the yaw/rear rotor)
- $K_{yp} = -0.0032$ (cross-torque thrust gain acting on yaw from pitch rotor)
- $K_{yy} = 0.0061$ (thrust force gain acting on the yaw axis from the yaw/rear rotor)
- $D_t = 0.1674$

6 Designing PD Controller

In this experiment, a proportional-derivative (PD) control is used to control the pitch and yaw axes to a desired angle.

6.1 Pitch and Yaw Models

The pitch and yaw can be modeled as separate transfer functions.

For Pitch:

$$\Theta(s) = \frac{K_{pp} D_t}{J_p s^2 + D_p s + K_{sp}} V_p \tag{7}$$

For Yaw:

$$\Psi(s) = \frac{K_{yy} D_t}{J_y s^2 + D_y s} V_y \tag{8}$$

7 Control Design

The PD control follows the structure:

$$u = K_p[r(t) - y(t)] - K_d \dot{y}(t) \tag{9}$$

Taking the Laplace transform:

$$Y(s) = \alpha[K_p(R(s) - Y(s)) - K_d s Y(s)] \tag{10}$$

$$\frac{Y(s)}{R(s)} = \frac{\alpha K_p}{1 + \alpha(K_p + K_d s)} \tag{11}$$

8 PD Controller for Pitch

For pitch,

$$\alpha = \frac{\Theta(s)}{V_p} = \frac{K_{pp} D_t}{J_p s^2 + D_p s + K_{sp}} \tag{12}$$

Thus, the closed-loop transfer function is:

$$\frac{Y(s)}{R(s)} = \frac{K_p D_t K_{pp}}{J_p s^2 + (D_p + K_{pp} D_t K_d)s + (K_{sp} + K_{pp} D_t K_p)/J_p} \tag{13}$$

Comparing with the standard second-order system:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{14}$$

We get:

$$\omega_n^2 = \frac{K_{sp} + K_{pp} D_t K_p}{J_p} \tag{15}$$

$$2\zeta\omega_n = \frac{D_p + K_{pp} D_t K_d}{J_p} \tag{16}$$

Solving for K_p and K_d:

$$K_p = \frac{J_p\omega_n^2 - K_{sp}}{K_{pp}D_t} \tag{17}$$

$$K_d = \frac{2\zeta\omega_n - D_p}{K_{pp}D_t} \tag{18}$$

9 PD Controller for Yaw

For yaw,

$$\alpha = \frac{K_{yy}D_t}{J_y s^2 + D_y s} \tag{19}$$

Thus, the closed-loop transfer function is:

$$\frac{Y(s)}{R(s)} = \frac{K_{yy}D_tK_p}{J_y s^2 + (D_y + K_{yy}D_tK_d)s + \frac{K_{yy}D_tK_p}{J_y}} \tag{20}$$

By comparing:

$$K_p = \frac{J_y\omega_n^2}{K_{yy}D_t} \tag{21}$$

$$K_d = \frac{2\zeta\omega_n J_y - D_y}{K_{yy}D_t} \tag{22}$$

10 Calculating ω_n and ζ using PO and Peak Time

The overshoot percentage and peak time are given by:

$$\%\ \text{Overshoot} = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \tag{23}$$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{24}$$

Given:

- % Overshoot = 5%
- $T_p = 2.5$ sec (for Pitch)
- $T_p = 3.5$ sec (for Yaw)

Solving, For Pitch:

$$\zeta = 0.6901, \quad \omega_n = 1.7363 \tag{25}$$

$$K_p = 116.7073 \quad K_d = 100.022 \tag{26}$$

For Yaw:

$$\zeta = 0.6901, \quad \omega_n = 1.7363 \tag{27}$$

$$K_p = 35.8373, \quad K_d = 38.0220 \tag{28}$$

11 Simulation Block Diagram

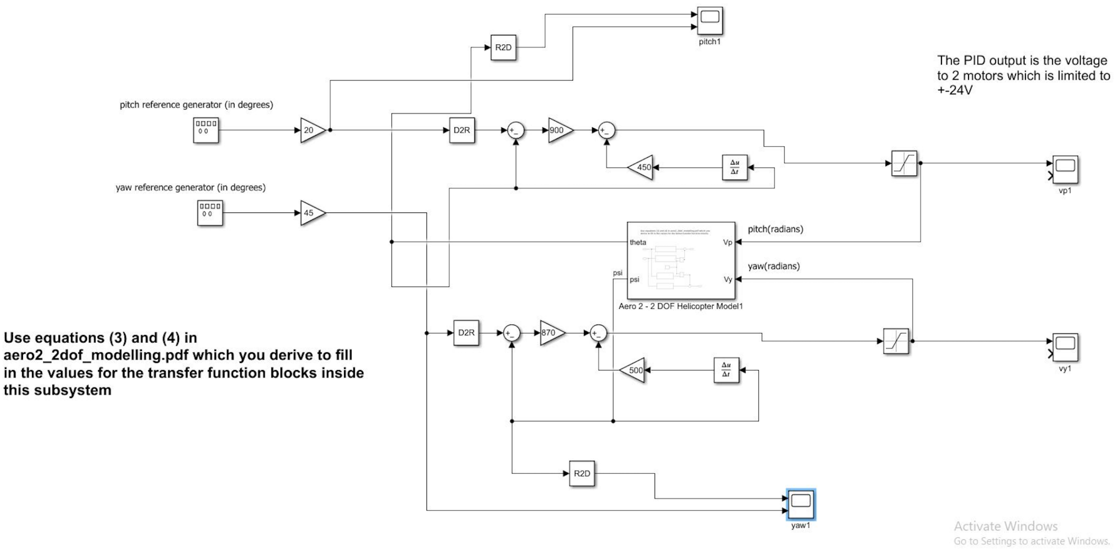

Figure 2: Simulation Block Diagram

12 Hardware Block Diagram

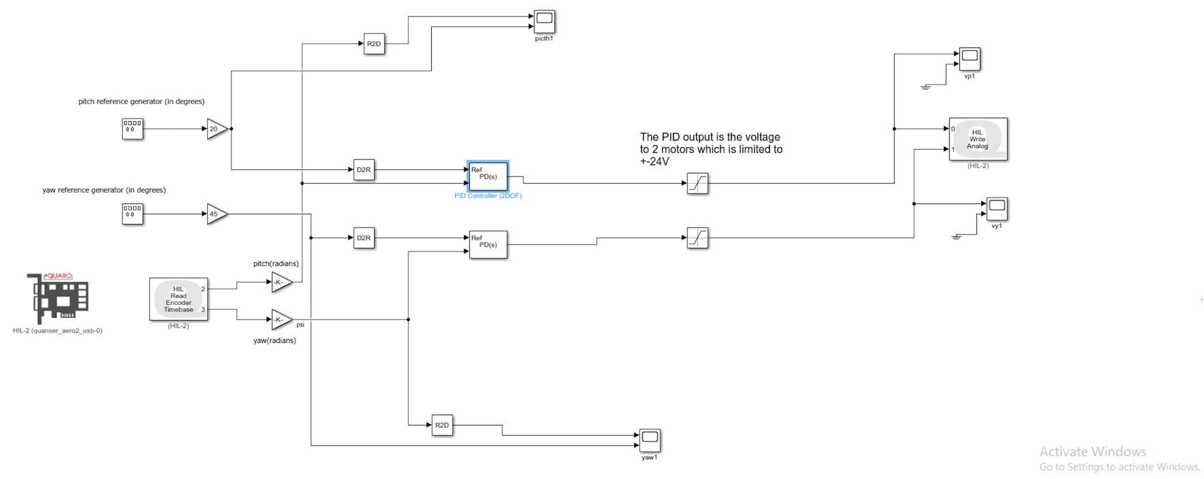

Figure 3: Hardware Block Diagram

13 Results

Hardware:

For Pitch:

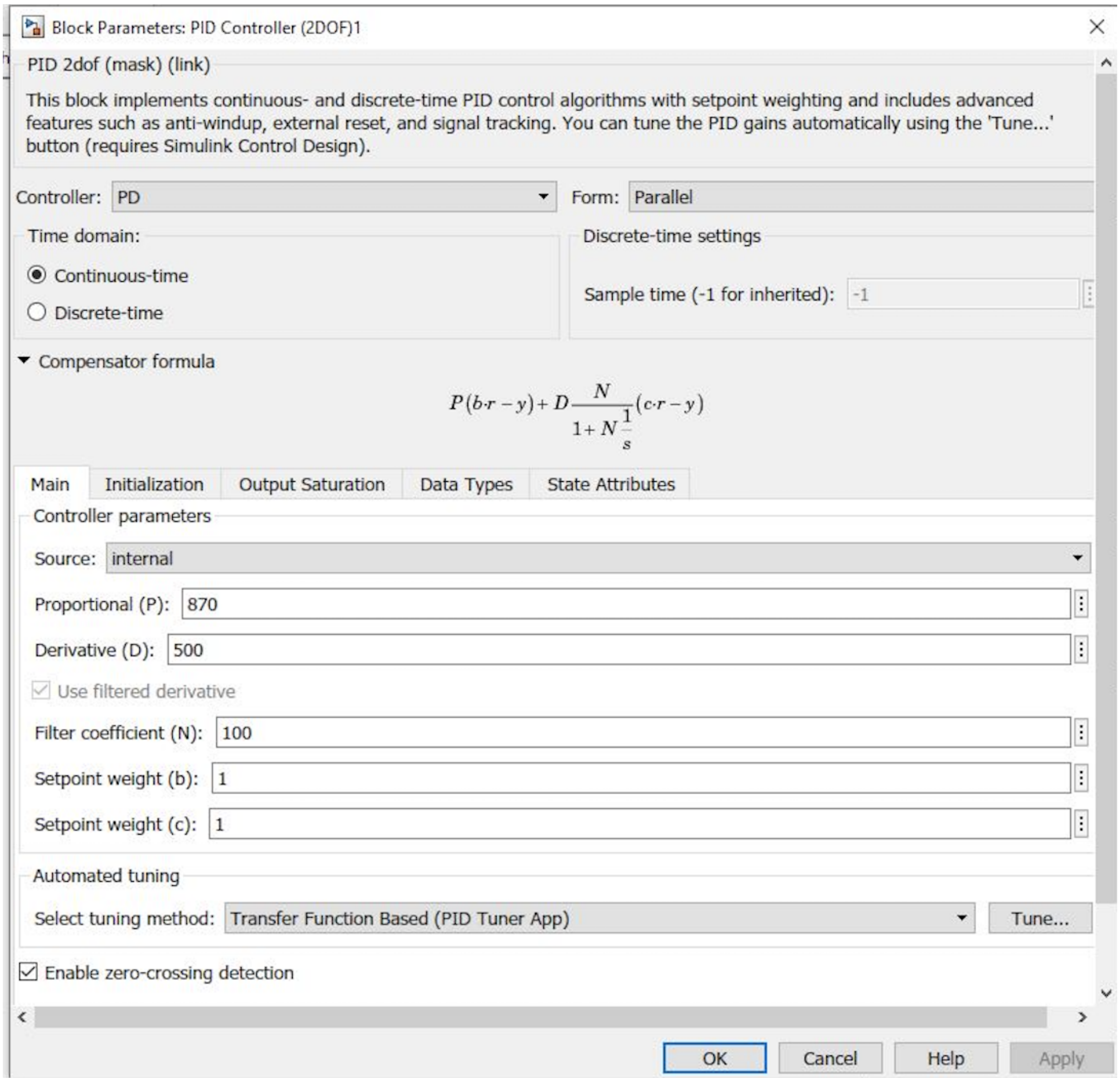

Figure 4: Pitch Parameters in Hardware

$$K_p = 920, \quad K_d = 480 \tag{29}$$

For Yaw:

$$K_p = 870, \quad K_d = 500 \tag{30}$$

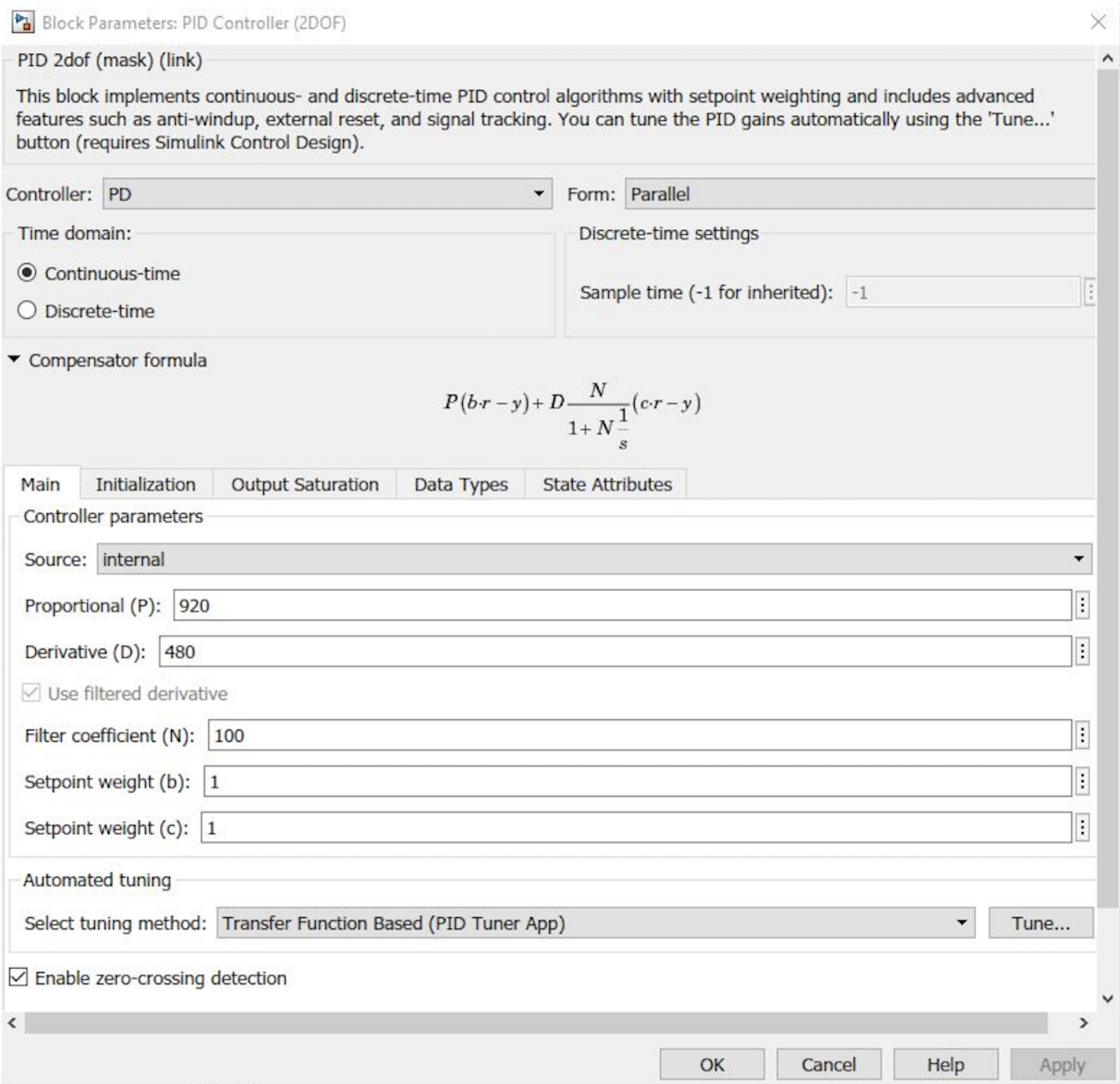

Figure 5: Yaw Parameters in Hardware

14 Challenges Faced

The main challenge was tuning. We were obtaining good results in simulation with certain parameter values, but the same performance was not observed in the hardware implementation. The smoothness of the response can be further improved by implementing an integrator.

15 Conclusion

The designed controllers successfully achieved the desired performance specifications. The results were validated through both simulation and hardware testing. Future improvements can include adaptive control techniques to enhance robustness.

Parameters	**Pitch**	**Yaw**
Peak Time (s)	0.934	2.8
Peak Overshoot (%)	3.1	3.9

Table 1: Simulation Response of Pitch and Yaw

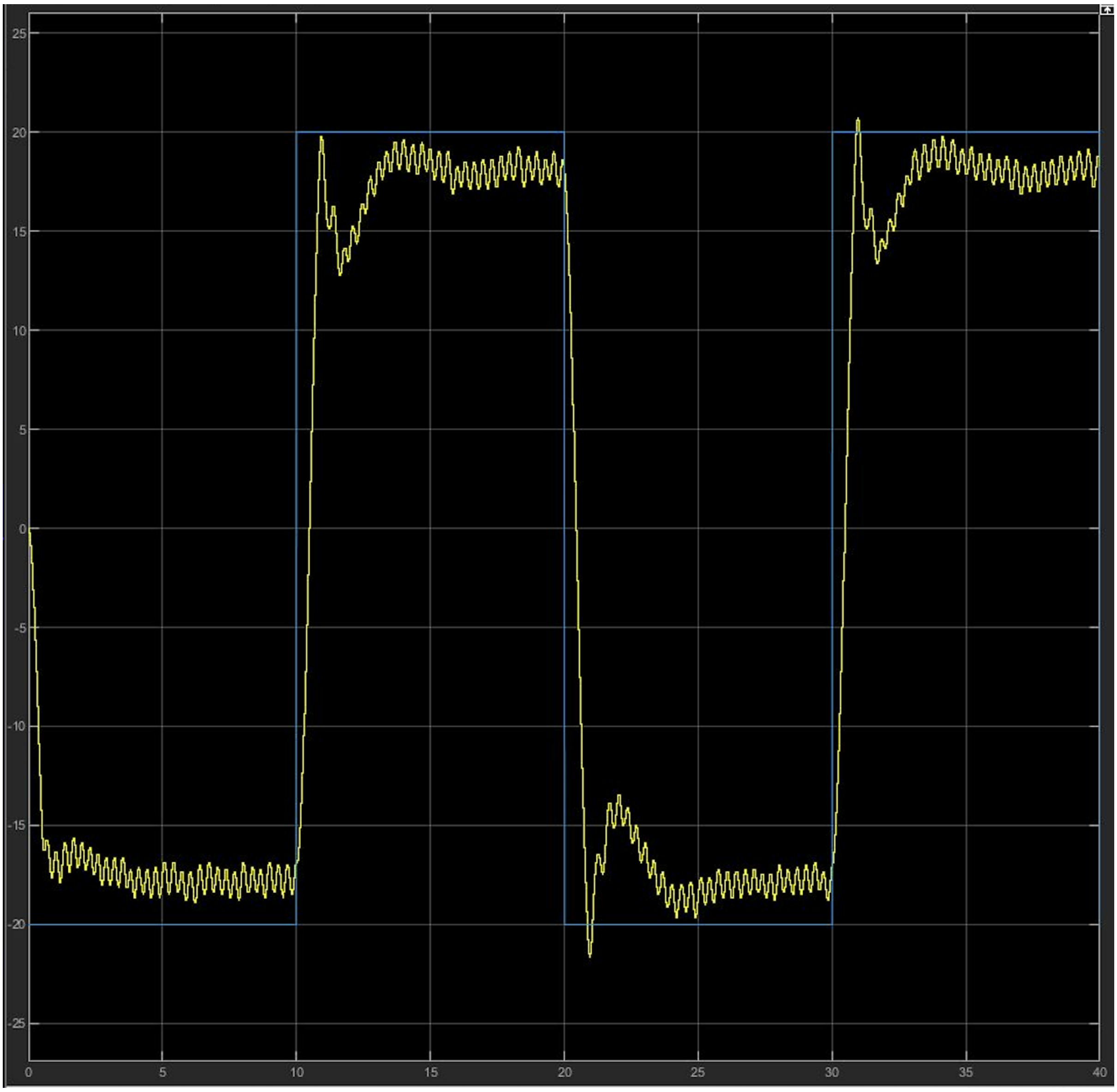

Figure 6: Pitch Response in Hardware

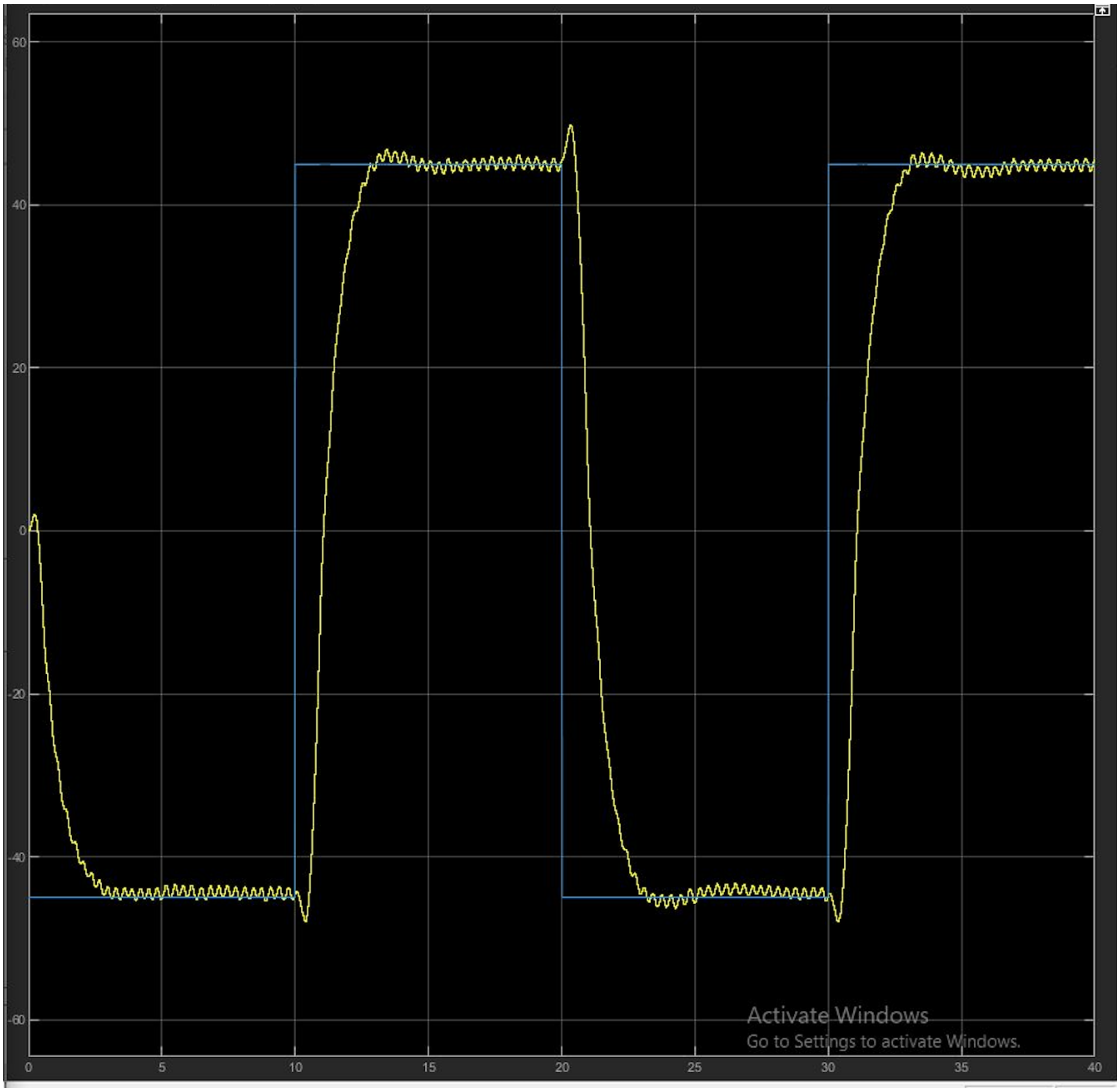

Figure 7: Yaw Response in Hardware

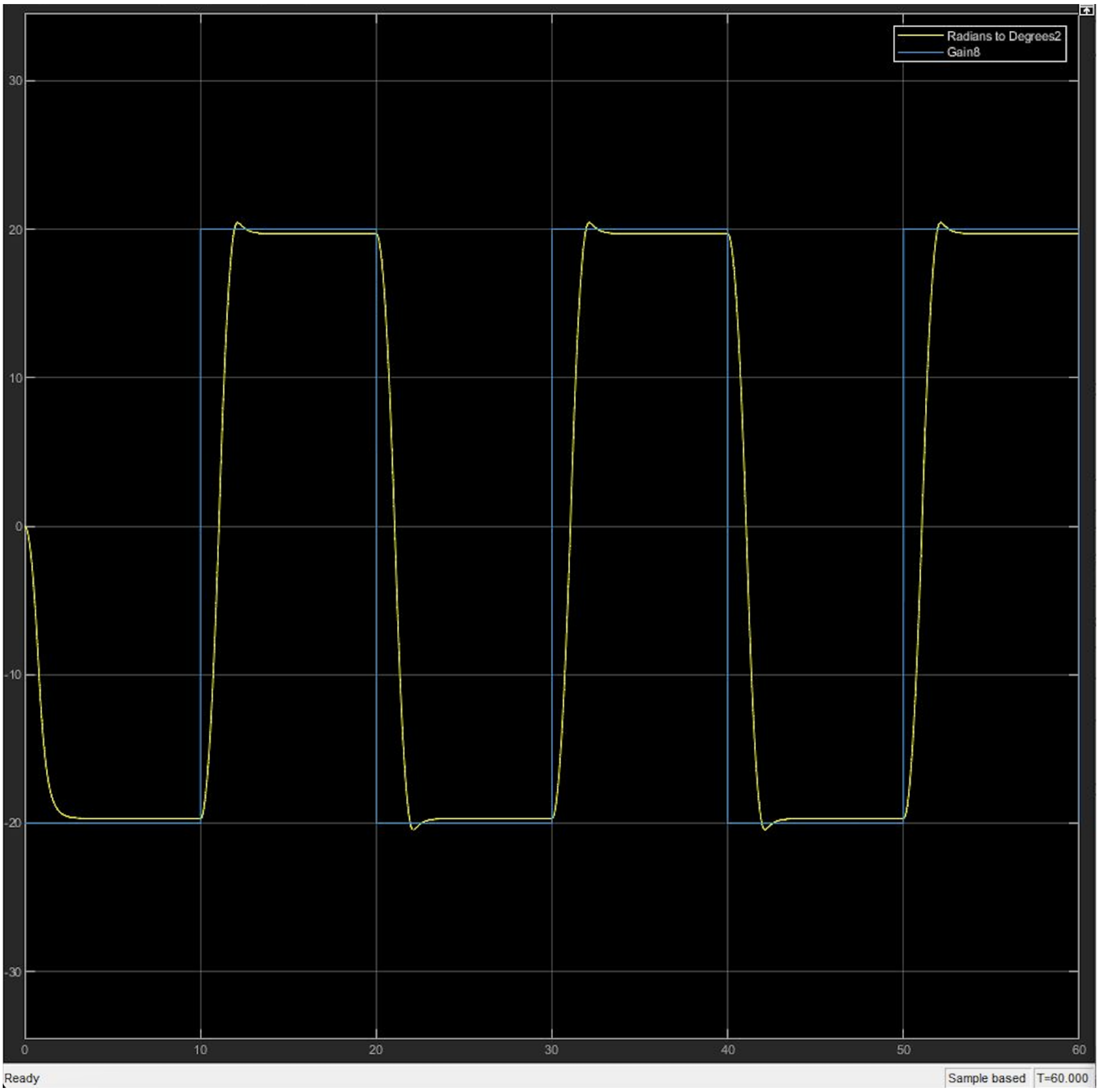

Figure 8: Pitch Response in Simulation

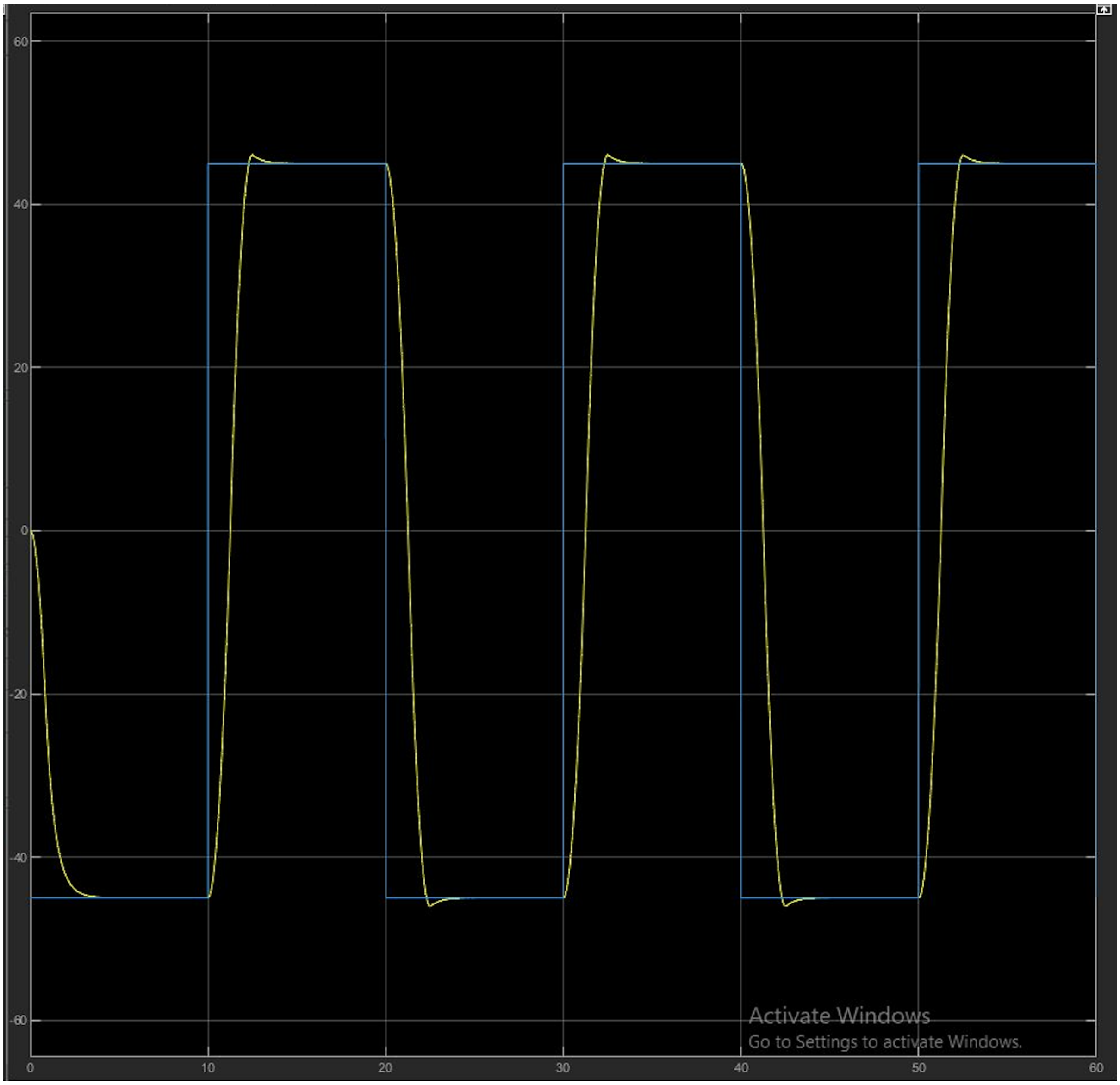

Figure 9: Yaw Response in Simulation